# Big Data intelligence Homework 1

Christoffer Brevik

October 24, 2024

## Introduction

This report explains my findings and the results of my code, implemented to solve the homework regarding the study of Social groups and the collective behaviors of their members. Each of the chapters describes one of the tasks given in the homework and will describe the python code I developed to solve the problem and detail all graphs and outputs of the code.

### Note about the programs

My code uses the *Pandas* library to import, group data and filter out columns not nessesary for each program. To plot data it uses the *pyplot* libray. Some of the programs also use special libraries to do calulations, these will be desribed in each chapter

# Task 1: ANOVA

## My code

My program first imports the excel sheet, group it by category and filters out all columns except category and average age in the groups. This data is then plotted to show the mean value for the average age value for the groups in each of the categories, as shown in Figure 1. Afterwards it implements the function *f_oneway* from the *scipy.stats* library to estimate a one-way ANOVA using an array of samples. Here we start with the null hypothesis stating that none of the categories can be distinguished by the inputted value (in this case *Average year*), and the resulting p-value describes whether or not this hyothesis is likely or not.

## Results

By running my program, the user first sees the following bar plot. This is shows the mean value for 'average year' for the groups in each category. Giving the user a figure to connect the result to.

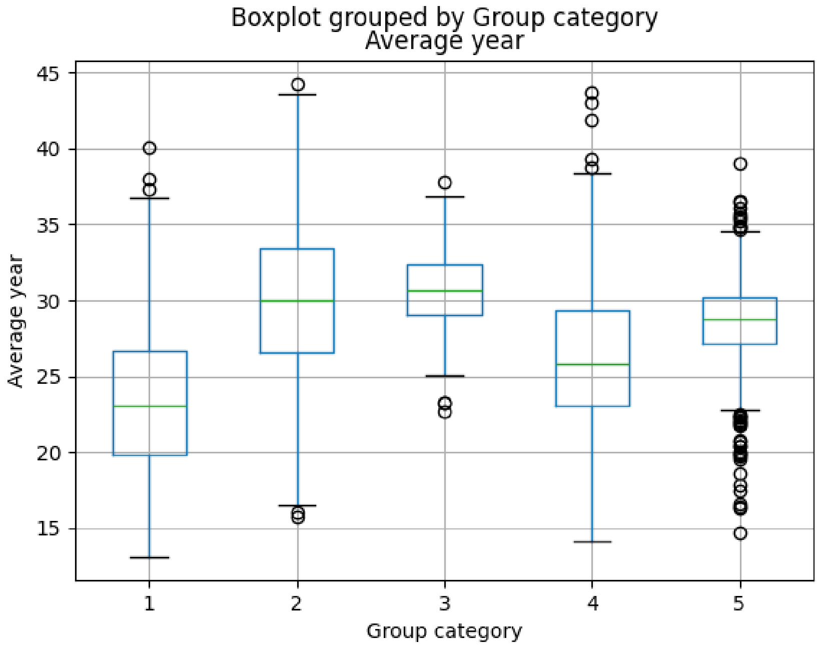

Figure 1: The bar plot produced by my code, showing the mean of the average age in each group for each category

After this figure is presented, the one-way ANOVA is calculated and we get the following output:

```
After doing a one-way ANOVA test, we get the following results:
The F-value of the one-way ANOVA test is:  171.50703270711966
The p-value of the one-way ANOVA test is:  1.0820916064752822e-126


The p-value is less than the treshold, we reject the null hypothesis
```

As the p-value is very low (being virtually zero) it is way lower than the threshold of 0,05 (or 5%). We can therefore discard the null hypothesis stating that none of the categories can be distinguished by the value *Average year*. This still does not give us any indication of how many of the categories or which category we can seperate, but this is not a part of the ANOVA process.

# Task 2a: Linear Regression

## My code

The program begins by loading data from an Excel sheet and filtering out any rows where the value ”Session number” is less than 20. It then performs linear regression for each dependent variable in relation to the independent variable, *Average year*. The dependent variables include *No-response ratio*, *Night-chat ratio* and *Picture ratio*.

For each dependent variable, a linear regression model is created, trained, and tested. The program computes the mean squared error to evaluate the model’s performance. These values are then printed to the user. Afterwards the program visualizes the results using subplots, where each plot represents the relationship between *Average year* and one of the dependent variables, along with the calculated regression line.

## Results

The program plots the following scatterplots with the corresponding regression line:

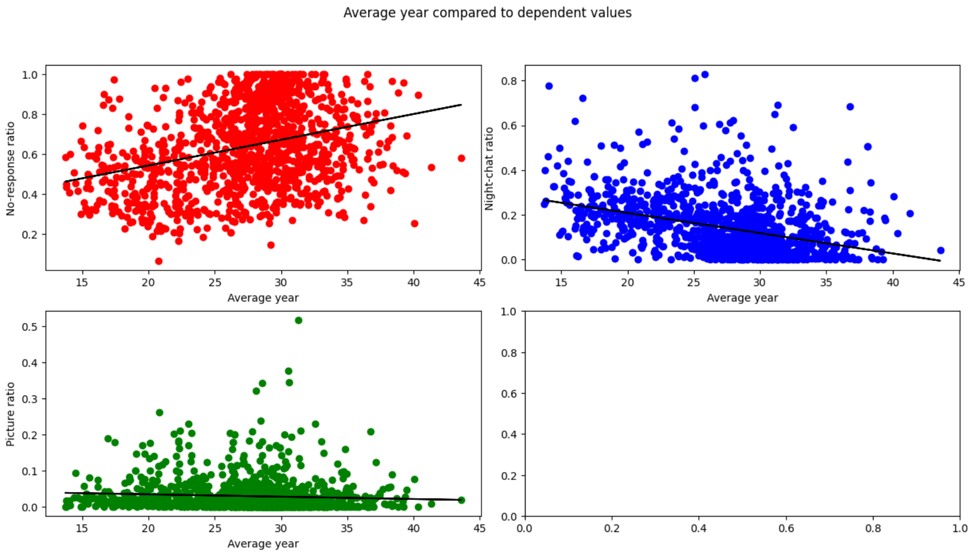

Figure 2: Scatterplots representing the dependent values as the y-axis and the independent value as the X-axis. The Regression line for each of the dependent values is also displayed

The different regression lines and their corresponding error are displayed in the console:

```
------------------------------------------------------------
Linear regression model for No-response ratio
Formula:            y = 0.01287x + 0.28629
Mean squared error: 0.03987
------------------------------------------------------------
Linear regression model for Night-chat ratio
Formula:            y = -0.00907x + 0.38993
Mean squared error: 0.01394
------------------------------------------------------------
Linear regression model for Picture ratio
Formula:            y = -0.00065x + 0.04738
Mean squared error: 0.00212
```

# Task 2b: Linear Functions

## My code

The program loads data from an Excel sheet, selecting columns 3 to 10 as features and columns 12 to 14 as the dependent variables. The ”Session number” column is used as weights for performing weighted linear regression.

For each of the three dependent variables, a weighted linear regression model is trained using the training data and session number as weights. The program outputs the regression equation and computes the Mean Squared Error for each model based on the test data.

The different regression lines and their corresponding mean squared errors are displayed in the console:

```
Linear regression model for Column 12:
Formula: y = -0.000*X3 + -0.000*X4 + -0.041*X5 + -0.021*X6 + 0.008*X7 + -0.002*X8
+ -0.042*X9 + -0.186*X10 + 0.480
Mean Squared Error for Column 12: 0.065

Linear regression model for Column 13:
Formula: y = -0.000*X3 + 0.000*X4 + 0.045*X5 + 0.039*X6 + -0.008*X7 + 0.011*X8
+ -0.043*X9 + 0.130*X10 + 0.280
Mean Squared Error for Column 13: 0.033

Linear regression model for Column 14:
Formula: y = -0.000*X3 + 0.000*X4 + 0.000*X5 + 0.024*X6 + -0.001*X7 + 0.000*X8
+ -0.012*X9 + -0.009*X10 + 0.044
Mean Squared Error for Column 14: 0.007
```

# Task 2c: Binary Classification

## My code

The program imports the data and filters out all elements not in category 1 or 4. We then split the data randomly into training (80% of data) and validation (20% of data) sets. We use the training data to train a multi-variable binary classification algorithm using all the columns, except group name and group category, as dependent values. After we are done training the algorithm, we test the algorithms accuracy by predicting the values in the test-set and estimating the MSE on these when compared to the actual values in the test data. The classifications accuracy is then displayed in the console.

Afterwards we estimate the optimal binary classification values by going trough all combinations of the columns (2-13). For each combination we create and train a binary classification algorithm, estimate the accuracy and finally we pick the best one. The best models accuracy and the columns used as dependent values are then shown to the users. This is the total console output after running the program

```
------------------------------------------------------------
Standard Logistic Regression Model
Accuracy: 0.7967032967032966

------------------------------------------------------------
Best Logistic Regression Model
Accuracy: 0.8571428571428571
Columns: (4, 6, 8, 9, 10)
```

Here *Standard Logistic Regression Model* is the model using all variables, whereas *Best Logistic Regression Model* was the model with the highest accuracy. As the sampling of data was random, I was worried that this was not consistent, but running the program similar times give very similar results with the same columns recommended by the program. Therefore, we see that the columns 4, 6, 8, 9 and 10 give the best results, with an accuracy of almost 86%.

## Task 3: Average of multiple linear regression

For this task i decided to redo the regression line for Night-chat ratio.

### My Code

My code is a modified version of the one used in 2a, where we will now split the dataset into 10 equal parts, perform individual linear regressions, and then average these out. My algorithm splits the data with three different strategies: **Random Sampling**, where data points are selected completely at random; **Stratified Sampling**, which divides the data into strata based on a specific characteristic and ensures representation from each stratum; and **Systematic Sampling**, which selects data points at regular intervals.

Important to note: I tested Stratefied for all the columns and found that the column 'Message number' gave the lowest errors for the given dataset. This is the one uses in my codes, and for the results below:

### Results

The program plots the following scatterplots with the corresponding regression line:

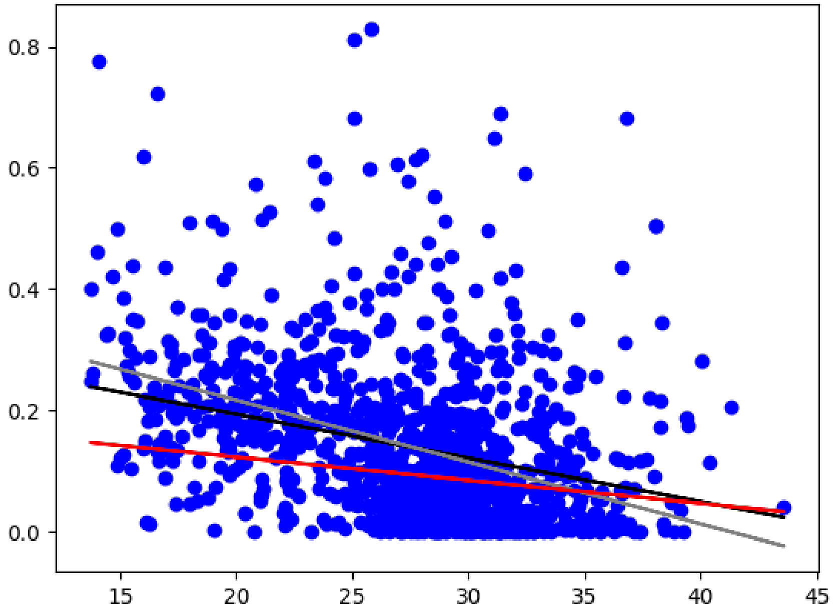

Figure 3: The average regression for the 'Night chat ratio' and it's 10 sub-regression-lines using the sampling stategies: Random (black), Stratified (gray) and Systematic (red)

The different regression lines and their corresponding mean squared are displayed in the console (Note I've replaced regression 1-8 by "..." to save space. Run the program to see the whole output):

```
------------------------------------------------------------
Random linear regression
Formula:            y = -0.00721x + 0.33879
Mean squared error: 0.01403
Plot color:         black
```

```
Statistics for Random sampling:
   Chunk     Mean  S. Deviation
0      1  0.18737       0.13135
1      2  0.21702       0.13584
2      3  0.15447       0.12664
3      4  0.10312       0.10837
4      5  0.13020       0.12027
5      6  0.17595       0.13745
6      7  0.12751       0.13708
7      8  0.09474       0.10127
8      9  0.10492       0.09961
9     10  0.11087       0.09573

------------------------------------------------------------
Stratified linear regression
Formula:            y = -0.01019x + 0.42146
Mean squared error: 0.01397
Plot color:         gray

Statistics for Stratified sampling:
   Chunk     Mean  S. Deviation
0      1  0.19719       0.16748
1      2  0.12999       0.14617
2      3  0.05365       0.06942
3      4  0.09623       0.07278
4      5  0.11955       0.10987
5      6  0.10505       0.07527
6      7  0.09991       0.10055
7      8  0.19897       0.11039
8      9  0.15115       0.07224
9     10  0.19863       0.10563

------------------------------------------------------------
Systematic linear regression
Formula:            y = -0.00382x + 0.20022
Mean squared error: 0.0167
Plot color:         red

Statistics for Systematic sampling:
   Chunk     Mean  S. Deviation
0      1  0.19529       0.10957
1      2  0.07895       0.00000
2      3  0.24242       0.00000
3      4  0.01256       0.00000
4      5  0.01989       0.00000
5      6  0.13208       0.00000
6      7  0.09225       0.00000
7      8  0.05747       0.00000
8      9  0.01818       0.00000
9     10  0.24432       0.00000
```

From the results we can see that for this dataset, a stratified sampling strategy based on the amount of messages in the group was the most beneficial. This strategy makes the algorithm almost as good as the original answer in 2a (which had the MSE of 0.01394, being lower by just 0.003). Randomly sampling the data was a little worse, with Systematic being clearly worse in this case. This may be due to the data not being ordered enough for this strategy to be beneficial.

Furthermore, the stratified linear regression seems to be almost equal to the one in 2a, with the other regressions having different graphs. I have drawn the following conclusions from my programs results:

- **Stratified sampling** consistently has moderate standard deviations across the 10 chunks. The deviations fluctuate, but they remain relatively balanced compared to the other strategies. This suggests that the data points within each chunk are more evenly distributed, leading to a more reliable representation of the overall dataset. This is also reflected on the lower MSE when compared to the other two strategies.
- **Random Sampling** has slightly higher fluctuations in the means and standard deviations across chunks. Though the variations aren't extreme, the lack of a structured approach like in stratified sampling makes it less reliable in terms of stability.
- **Systematic Sampling** shows significant instability, with many chunks having a standard deviation of 0, indicating that all the values within those chunks are the same. This can lead to biased results because the method selects data at fixed intervals, potentially missing important variations in the dataset.

Based on the results presented, Stratified sampling appears to be both the most stable and beneficial sampling strategy.